

BEST AVAILABLE COPY

AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions:

Claims 1-24. (Cancelled)

25. (New) A method comprising:

receiving an attribute at a particular node in a network, said attribute being disseminated among a plurality of nodes comprising the network;

determining if the attribute appears caught in a loop in the network; and

registering the attribute for the node if the attribute does not appear to be caught in a loop in the network.

26. (New) The method of claim 25 wherein determining if the attribute appears caught in a loop in the network comprises:

comparing an index key of the attribute to a plurality of previously stored index keys of attributes previously registered for the node; and

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop.

27. (New) The method of claim 25 wherein the attribute comprises an index key, a value associated with the index key, and an incarnation identifier for the value, and wherein determining if the attribute appears caught in a loop in the network comprises:

comparing the index key of the attribute to a plurality of previously stored index keys of attributes previously registered for the node;

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, comparing the incarnation identifier of the attribute to a previously stored incarnation identifier of the corresponding previously stored attribute; and

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does not match the previously stored incarnation identifier, proceeding as if the attribute is not caught in a loop.

28. (New) The method of claim 25 wherein the attribute comprises an index key, a value associated with the index key, and an incarnation identifier for the value, and wherein determining if the attribute appears caught in a loop in the network comprises:

comparing the index key of the attribute to a plurality of previously stored index keys of attributes previously registered for the node;

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, comparing the incarnation identifier of the attribute to a previously stored incarnation identifier of the corresponding previously stored attribute;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does not match the previously stored incarnation identifier, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does match the previously stored incarnation identifier, comparing a receiving port identifier of the attribute to a previously stored receiving port identifier of the corresponding previously stored attribute; and

if the index key is among the previously stored index keys, the incarnation identifier of the attribute does match the previously stored incarnation identifier, and the receiving port identifier of the attribute does match the previously stored receiving port identifier, proceeding as if the attribute is not caught in a loop.

29. (New) The method of claim 25 wherein the attribute comprises an index key, a value associated with the index key, and an incarnation identifier for the value, and wherein determining if the attribute appears caught in a loop in the network comprises:

comparing the index key of the attribute to a plurality of previously stored index keys of attributes previously registered for the node;

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, comparing the incarnation identifier of the attribute to a previously stored incarnation identifier of the corresponding previously stored attribute;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does not match the previously stored incarnation identifier, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does match the previously stored incarnation identifier, comparing a receiving port identifier of the attribute to a previously stored receiving port identifier of the corresponding previously stored attribute;

if the index key is among the previously stored index keys, the incarnation identifier of the attribute does match the previously stored incarnation identifier, and the receiving port identifier of the attribute does match the previously stored receiving port identifier, proceeding as if the attribute is not caught in a loop; and

if the index key is among the previously stored index keys, the incarnation identifier of the attribute does match the previously stored incarnation identifier, and the receiving port identifier of the attribute does not match the previously stored receiving port identifier, proceeding as if the attribute is caught in a loop.

30. (New) The method of claim 25 wherein registering the attribute comprises:
storing the attribute to local memory.

31. (New) The method of claim 25 wherein storing the attribute to local memory comprises at least one of:

overwriting an older version of the attribute in the local memory; and
recording a receiving port identifier for the attribute.

32. (New) The method of claim 25 further comprising:

ignoring the attribute if the attribute appears to be caught in a loop.

33. (New) The method of claim 25 further comprising:

multicasting the attribute from each port of the node except a port at which the attribute was received if the attribute appears not to be caught in a loop.

34. (New) A machine readable medium having machine executable instructions stored thereon, the execution of which to implement a method comprising:

receiving an attribute at a particular node in a network, said attribute being disseminated among a plurality of nodes comprising the network;

determining if the attribute appears caught in a loop in the network; and

registering the attribute for the node if the attribute does not appear to be caught in a loop in the network.

35. (New) The machine readable medium of claim 34 wherein determining if the attribute appears caught in a loop in the network comprises:

comparing an index key of the attribute to a plurality of previously stored index keys of attributes previously registered for the node; and

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop.

36. (New) The machine readable medium of claim 34 wherein the attribute comprises an index key, a value associated with the index key, and an incarnation identifier for the value, and wherein determining if the attribute appears caught in a loop in the network comprises:

comparing the index key of the attribute to a plurality of previously stored index keys of attributes previously registered for the node;

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, comparing the incarnation identifier of the attribute to a previously stored incarnation identifier of the corresponding previously stored attribute; and

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does not match the previously stored incarnation identifier, proceeding as if the attribute is not caught in a loop.

37. (New) The machine readable medium of claim 34 wherein the attribute comprises an index key, a value associated with the index key, and an incarnation identifier for the value, and wherein determining if the attribute appears caught in a loop in the network comprises:

comparing the index key of the attribute to a plurality of previously stored index keys of attributes previously registered for the node;

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, comparing the incarnation identifier of the attribute to a previously stored incarnation identifier of the corresponding previously stored attribute;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does not match the previously stored incarnation identifier, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does match the previously stored incarnation identifier, comparing a receiving port identifier of the attribute to a previously stored receiving port identifier of the corresponding previously stored attribute; and

if the index key is among the previously stored index keys, the incarnation identifier of the attribute does match the previously stored incarnation identifier, and the receiving port identifier of the attribute does match the previously stored receiving port identifier, proceeding as if the attribute is not caught in a loop.

38. (New) The machine readable medium of claim 34 wherein the attribute comprises an index key, a value associated with the index key, and an incarnation identifier for the value, and wherein determining if the attribute appears caught in a loop in the network comprises:

comparing the index key of the attribute to a plurality of previously stored index keys of attributes previously registered at the node;

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, comparing the incarnation identifier of the attribute to a previously stored incarnation identifier of the corresponding previously stored attribute;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does not match the previously stored incarnation identifier, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does match the previously stored incarnation identifier, comparing a receiving port identifier of the attribute to a previously stored receiving port identifier of the corresponding previously stored attribute;

if the index key is among the previously stored index keys, the incarnation identifier of the attribute does match the previously stored incarnation identifier, and the receiving port identifier of the attribute does match the previously stored receiving port identifier, proceeding as if the attribute is not caught in a loop; and

if the index key is among the previously stored index keys, the incarnation identifier of the attribute does match the previously stored incarnation identifier, and the receiving port identifier of the attribute does not match the previously stored receiving port identifier, proceeding as if the attribute is caught in a loop.

39. (New) The machine readable medium of claim 34 wherein registering the attribute comprises:

storing the attribute to local memory.

40. (New) The machine readable medium of claim 34, the method further comprising:

ignoring the attribute if the attribute appears to be caught in a loop.

41. (New) The machine readable medium of claim 34, the method further comprising:

multicasting the attribute from each port of the node except a port at which the attribute was received if the attribute appears not to be caught in a loop.

42. (New) A system comprising:

a processor; and

a machine readable medium accessible by the processor, said machine readable medium to store instructions that when executed by the processor implement

receiving an attribute at a particular node in a network, said attribute being disseminated among a plurality of nodes comprising the network;

determining if the attribute appears caught in a loop in the network; and

registering the attribute for the node if the attribute does not appear to be caught in a loop in the network.

43. (New) The system of claim 42 wherein the attribute comprises an index key, a value associated with the index key, and an incarnation identifier for the value, and wherein determining if the attribute appears caught in a loop in the network comprises:

comparing the index key of the attribute to a plurality of previously stored index keys of attributes previously registered for the node;

if the index key is not among the previously stored index keys, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, comparing the incarnation identifier of the attribute to a previously stored incarnation identifier of the corresponding previously stored attribute;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does not match the previously stored incarnation identifier, proceeding as if the attribute is not caught in a loop;

if the index key is among the previously stored index keys, and the incarnation identifier of the attribute does match the previously stored incarnation identifier, comparing a receiving port identifier of the attribute to a previously stored receiving port identifier of the corresponding previously stored attribute;

if the index key is among the previously stored index keys, the incarnation identifier of the attribute does match the previously stored incarnation identifier, and the receiving port identifier of the attribute does match the previously stored receiving port identifier, proceeding as if the attribute is not caught in a loop; and

if the index key is among the previously stored index keys, the incarnation identifier of the attribute does match the previously stored incarnation identifier, and the receiving port identifier of the attribute does not match the previously stored receiving port identifier, proceeding as if the attribute is caught in a loop.

44. (New) The system of claim 42 wherein registering the attribute comprises:
storing the attribute to local memory.

45. (New) The system of claim 42, the method further comprising:
ignoring the attribute if the attribute appears to be caught in a loop.

46. (New) The system of claim 42, the method further comprising:

multicasting the attribute from each port of the node except a port at which the attribute was received if the attribute appears not to be caught in a loop.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.